Tenemos un RDD con información sobre tests de covid19 (fecha, dni, id_localidad, resultado)

otro RDD con la información de las localidades (id_localidad, nombre, provincia)

Utilizando el API de RDD de PySpark queremos: a) Indicar las provincias que hayan incrementado más del 20% la cantidad de tests mensuales entre algún mes del primer trimestre del 2021. Esto es, que hayan aumentado 20% de Enero a Febrero o aumentado 20% de Febrero a Marzo.

```
#perdon la desprolijidad, di mil vueltas y me quede corto de tiempo como para presentar el codigo mejor, tampoco pude dar el
#en el punto b, solo la id.

#lo mismo con los nombres de las variables, me quedaron horrible.
#podria haber reutilizado junto a cache algunas variables pero no me dio el tiempo para acomodarlo
#la forma en la que obtuve la cantidad de localidades y tests realizados es fea
```

```
test_covid_rdd = sc.parallelize(tests_covid)
localidades_rdd = sc.parallelize(localidades)
```

```
def funcion_trimestre(reg):
  if '2021-01' in reg[0]:
    return (1,0,0)
  elif '2021-02' in reg[0]:
    return (0,1,0)
  elif '2021-03' in reg[0]:
    return (0,0,1)
  else:
    return 0

por_trimestre = test_covid_rdd.map(lambda x: ((x[2], funcion_trimestre(x))))
localidades_rdd = localidades_rdd.map(lambda x: (x[0], x[2]))
test_localidad = por_trimestre.join(localidades_rdd).map(lambda x: (x[1][1], (x[1][0][0], x[1][0][1], x[1][0][2])))


#considero que no hay provincias que vacunen sin antes anotarlos (en tests_covid) ni hay vacunados sin asignarle una provinc

suma = test_localidad.reduceByKey(lambda a,b: ((a[0] + b[0], a[1] + b[1], a[2] + b[2])))
  # me quedo con las que aumentaron un 20 porciento
suma.filter(lambda x: (abs(x[1][1] - x[1][0])/x[1][0] > 0.2) | abs(x[1][2] - x[1][1])/x[1][1] > 0.2)
```

```
# punto b
localidades_rdd = localidades_rdd.map(lambda x: (x[0], x[1]))

test_covid_primer_t = test_covid_rdd.filter(lambda x: ['01-2021', '02-2021', '03-2021' in x[0]])

test_totales = test_covid_rdd.map(lambda x: 1).reduce(lambda a,b: a+b)

test_covid_localidad = test_covid_rdd.map(lambda x: (x[2], (1,1 if x[3]==1 else 0)))\
    .reduceByKey(lambda a,b: (a[0]+b[0], a[1]+b[1])).cache()


localidades_totales = localidades_rdd.count()

test_covid_localidad.filter(lambda x: int(x[1][0]) > int(test_totales/localidades_totales))\
      .reduce(lambda a,b: a if a[1][1]/a[1][0] > b[1][1]/b[1][0] else b)
```

0 s completado a las 20:19